

Software Requirements Specification

一、引言

1.1 需求规格说明书编写目的

本软件需求规格说明书对软件需求工程课程项目《开源项目分析平台》的实现做了充分用户需求分析，完善并扩展其功能，美化其界面。清晰地描述了开源项目分析平台的各项需求，包括功能性需求和非功能性需求。

本软件需求规格说明书旨在明确我们需要开发的开源项目分析平台应该具有的各项基本要求，帮助开发者通过单一平台对不同托管平台上的开源项目进行项目分析、比较，以更好的进行项目管理。使得项目经理和软件开发人员以及系统分析师可以清楚地了解不同用户的不同需求，为后续的系统设计和软件开发打下坚实的文档基础。

1.2 软件项目背景

本项目是浙江大学 2022-2023 学年秋冬学期的《软件需求工程》课程项目，目标为优化一个开源项目分析平台。在其项目信息检索模块，更新频率折线图模块，贡献比例饼状图模块，issues 柱状图模块这四个已经实现的模块下进行一系列优化处理。

1.3 目标读者和阅读建议

1.3.1 软件需求规格说明书的组成部分

本软件需求规格说明书主要由如下几个部分构成：总体描述、系统需求分析概述、功能需求、数据流图、业务规则和业务算法以及附录这些部分。

1.3.2 软件需求规格说明书面向的读者对象

项目经理，设计人员，程序开发人员，销售人员，测试人员，用户，其他人员（比如部门领导、公司领导、政府机关领导）

在阅读本软件需求规格说明书的时候，首先要了解产品的功能概貌，然后可以根据自身的需要对每一功能进行实当的了解。

1.4 参考文献与资料

《软件设计文档国家标准》《软件工程项目开发文档范例》

《软件需求》（刘伟琴、刘洪涛译）《软件工程——实践者的研究方法》

二、总体描述

2.1 产品基本信息

产品名称：开源项目分析平台

任务提出者：浙江大学软件需求工程课程任课老师：万志远，林海

开发团队：软件需求工程 xxyyz

用户类型：分析者

系统基本功能：统计一段时间不同贡献者的各类贡献数据，对 PyTorch 项目进行数据可视化，横向多个项目，纵向同一项目对比，实现优化数据获取。

性能：允许 10 个人同时进行项目的分析比较。

完成期限：本学期期末验收之前。

2.2 产品前景

随着信息技术的不断发展，开源项目的数量和规模都在不断增加，而如何让这些数据有规则地被整理，以及明确贡献者的具体贡献就成了亟待解决的问题。

2.3 用户类别与特征

分析者 可以进入开源项目分析平台，查看 Pytorch 项目的一系列信息，并与 pandas 项目进行比较。

2.4 设计和实现上的约束

本平台仅用于开源项目 Pytorch 的分析，涉及的是项目信息的搜索职能，**不包含项目本身的职能**，如涉及具体项目的项目成果，在具体设计和实现上按照以下约束进行。

2.4.1 网络并发量与性能

允许 10 个人同时进行项目的分析比较。

2.4.2 数据存储与安全保证

项目产品使用 MySQL 数据库作为数据库引擎，按照数据的产生、转换和存储的策略，通过将数据导入数据库的方式进行数据的存储操作。同时我们也需要保证存储数据的**完整性，保密性和可用性**。

数据的完整性要求数据不能被随意修改。

保密性要求数据不能被权限不足的人进行访问。

可用性要求做到避免因为数据泄露使得合法使用者无法接触可用数据。

2.5 开发、测试和运行环境

开发环境：Windows 11，Ubuntu 20.04，macOS Ventura 13.1

开发框架：前端 Vue+JavaScript，后端 django 3.0+版本

测试环境：可以联网的电脑，阿里云服务器，相关的开源测试工具，如 Junit

运行环境：可以联网的电脑，浏览器(尽量适配 IE 浏览器)

2.6 用户文档

本产品交付时将为用户提供三类文档，分别是描述类文档、过程类文档、参考类文档，主要帮助用户可以快速入门开源项目分析平台的使用，并在遇到实际问题时可以通过文档查阅快速解决所遇到的问题。

2.7 产品开发项目设计人员

项目总负责人 陈奕宇：统筹整个项目工程的实现，领导项目小组完成全部项目工作内容。

前端开发 江小小：建立系统框架；数据库设计；概要设计；参加技术评审。

后端开发 黄亦霄：负责进行编码工作与单元测试，进行系统集成，及时解决测试时出现的问题。

测试工作 陈思全：编写测试方案和测试用例，进行系统测试，向开发人员反馈 BUG。

文档工作 姚逸飞：记录项目流程；撰写项目说明书；组织编写项目计划和试方案。

2.8 系统需求分析概述

需求追踪

用例编号	用例名称	修改原因
FE-1	数据源缓存	实现针对数据源的缓存，将数据与实际数据来源解耦，降低由于网络问题可能引发的风险
FE-2	数据过滤	细化数据过滤，对可能缺失的数据进行特殊处理，提升用户友好程度
FE-3	项目历史数据访问	优化从 GitHub 上获取信息的方式，在 GitHub 接口不可用时仍能提供项目历史数据访问
FE-4	统计核心贡献者	统计一段时间内来自不同贡献者的代码提交数，指出项目的核心贡献者
FE-5	筛选相关讨论	取得 commit 信息，筛选出设计相关的讨论
FE-6	可视化展示讨论数量	以周为单位，可视化展示设计讨论数量随时间的变化
FE-7	可视化占比	以 PyTorch 项目为例，进行数据可视化。具体 1. 查询 stargazer, committer, issue 信息，按 company 组织数据 2. 计算占比并将其可视化，在可视化效果上体现出占比大小等不同 3. 进一步，按内部不同模块进行上述处理
FE-8	衡量代码设计质量	定义 code smell 并衡量最新版代码设计质量
FE-9	实现 URL 跳转	完善系统功能

需求优先级分析

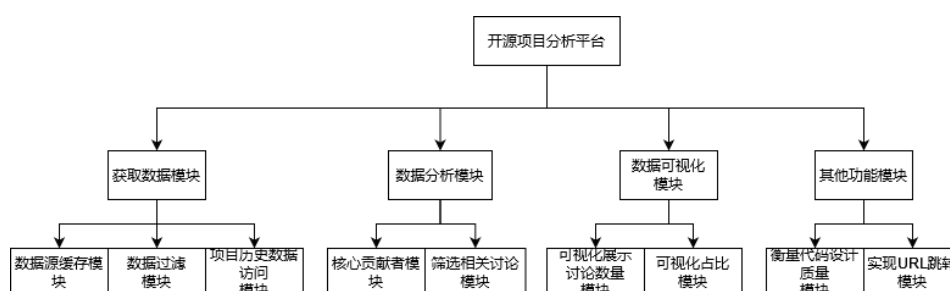
本需求优先级分析模块依据风险-价值四象限矩阵进行评级，即按照下表所示四象限对需求进行优先级分类。

	高价值	低价值
高风险	首先处理	避免
低风险	其次处理	最后处理

其中，用例包括统计、可视化、获取、项目重构。

用例编号	用例名称	风险	价值	优先级
FE-1	数据源缓存	低	中	中
FE-2	数据过滤	低	中	中
FE-3	项目历史数据访问	低	中	中
FE-4	统计核心贡献者	低	高	高
FE-5	筛选相关讨论	低	高	高
FE-6	可视化展示讨论数量	低	高	高
FE-7	可视化占比	低	高	高
FE-8	衡量代码设计质量	低	高	高
FE-9	实现 URL 跳转	低	高	高

系统模块架构图



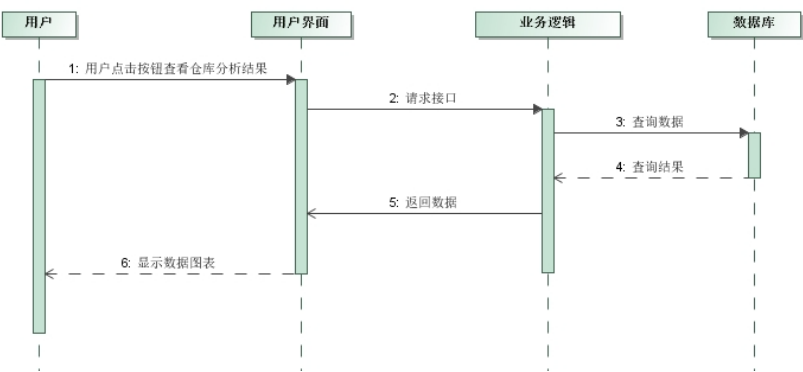
三、功能需求

3.1 显示

3.1.1 use case

用例编号	SE-US-02	用例名称	显示数据
创建人	江小小	最后修改人	江小小
创建日期	2022/11/9	最后修改日期	2022/11/9
角色	开源代码分析者	需求来源	以开源代码分析者为代表的用户
主要参与者		代码分析者	
描述		用户通过按下界面上的查看仓库分析结果按钮，获取项目相关分析信息	
前置条件		分析平台运行正常、用户使用设备运行正常 与数据接口连接畅通	
触发器		用户按下查看按钮	
后置条件		无	
主干过程		用户进入代码分析平台界面、 用户按下查看按钮、界面显示显示项目信息	
分支过程		无	
异常		页面出现异常（发生于主干过程步骤 1） 处理：根据网站联系方式联系系统管理员 系统结束用例	
优先级		最高	
使用频率		高（预估约为每日平台使用者数量）	
假设		无异常	
备注		无	

3.1.2 sequence diagram



3.1.3 activity diagram

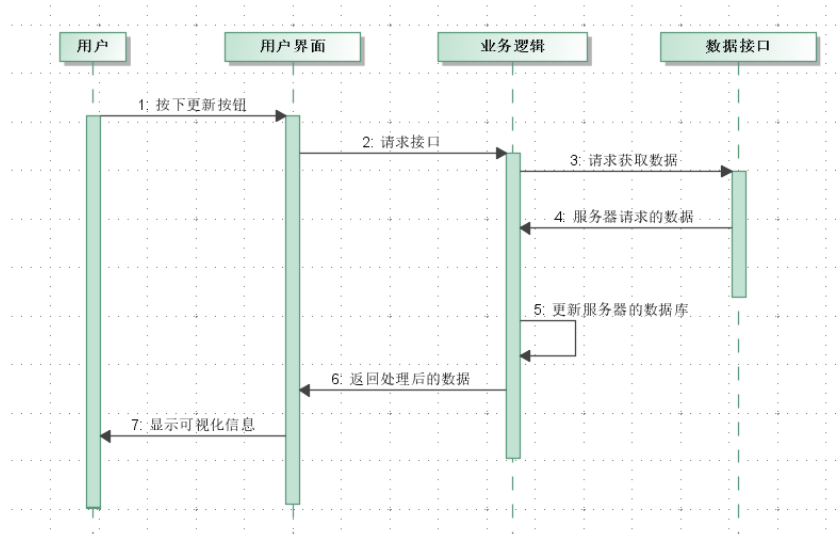


3.2 更新数据

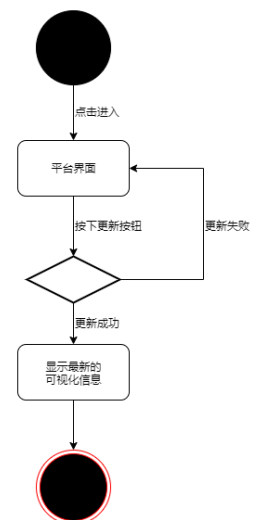
3.2.1 use case

用例编号	SE-US-01	用例名称	更新数据
创建人	陈奕宇	最后修改人	陈奕宇
创建日期	2022/11/8	最后修改日期	2022/11/8
角色	开源代码分析者	需求来源	以开源代码分析者为代表的用户
主要参与者	代码分析者		
描述	用户通过按下界面上的更新按钮，获取最新的项目相关分析信息		
前置条件	分析平台运行正常、用户使用设备运行正常、与数据接口连接畅通		
触发器	用户按下更新按钮		
后置条件	数据库更新项目最新的相关信息		
主干过程	用户进入代码分析平台界面 用户按下更新按钮 3. 完成更新，重新显示项目的最新信息		
分支过程	无		
异常	页面出现异常（发生于主干过程步骤 1） 处理：根据网站联系方式联系系统管理员 系统结束用例 更新时浏览器显示数据获取速度限制异常信息（发生于主干过程步骤 3） 处理：更新数据量较大，遭受数据接口限制，等待一段时间后重试 系统结束用例		
优先级	最高		
使用频率	高（预估约为每日平台使用者数量）		
假设	无异常		
备注	无		

3.2.2 sequence diagram



3.2.3 activity diagram



四、业务逻辑与算法

4.1 业务逻辑

社区成员信息获取和分析

本项目定义的社区成员由 `issue`、`commit` 的提出者和为在项目页面点起 `star` 表示赞赏的 `stargazer`。三类成员分别代表了项目前景、项目生产力水平和项目影响三个方面的活跃度，响应了系统着眼“项目社区活跃度”的业务需求。对此三方面的成员，从宏观上分析其数量变化趋势、按公司聚类的数量及其变化趋势。

数量变化趋势展示方式为折线图，以日为最小统计单元，用户可以调整时间窗口大小为周、月、半年、年，并在时间轴中滑动时间窗口以获取对应图线。

按公司聚类展示方式为按公司名称关键词出现次数生成的词云和按公司全名聚类的饼图，统计时间跨度与数量变化趋势中选择的时间窗口一致。



设计讨论信息获取和分析

本项目定义的设计讨论包括在下图的调查中得分最高的前三项，即“可维护性”（包括项目未来规划、操作系统平台支持和代码标准等）、“健壮性”（还包括安全性）、“代码实现”（即 `issue` 的具体解决）。

本项目定义主要贡献者，为在选定的时间窗口内，按 commit 数从高到低排序，占人数前 10% 的开发者。

设计讨论数量趋势展示方式为累加折线图，展示选定时间窗口内由分别由主要贡献者和其他贡献者贡献的 commit 数量。

设计讨论关键词展示方式为词云，展示由 commit 信息中提取得关键词的聚类统计。

Category	Occurrences	Description
	Agreed on	
code	11	Implementation issues
maintainability	14	Future plans, OS support, code standards...
testing	1	Tests and testability
robustness	13	Robustness, safety, security
performance	2	Performance, runtime optimization
configuration	4	Configuration files, flags and options
documentation	1	Documentation in-code and off-code
clarification	7	Generic question

4.2 业务算法

信息获取

信息获取行为的触发器是用户要求、开发人员后台控制和时钟控制。对于社区成员信息，获取方式是 GitHub API。对于 commit 信息，获取方式是更新本地项目镜像并使用 `git log` 命令生成的 commit log 文件。

信息存储

信息存储方式为数据库表本地存储，json 文件完成迁移、前后端通信。数据库表参见附录 ER 图。

信息更新

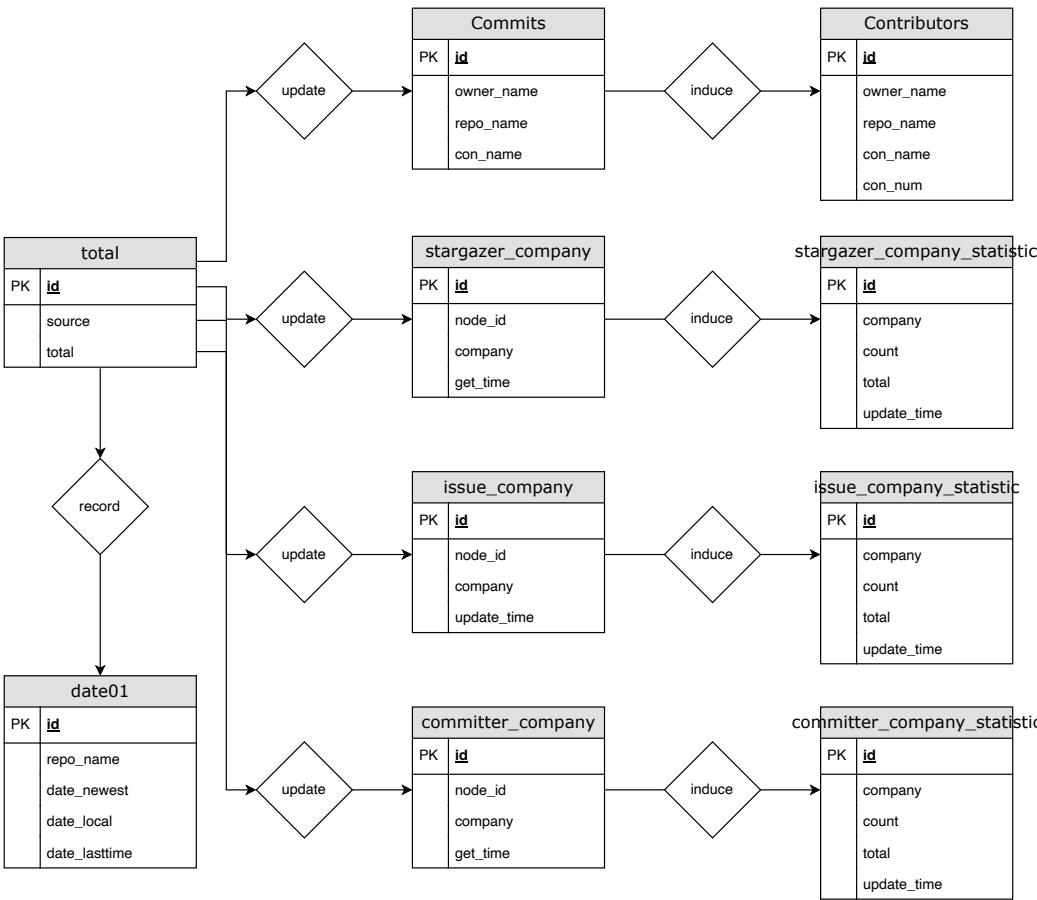
信息更新在信息获取同时进行。由于全部信息属于增量型历史记录,旧记录不会被删除,故更新时自记录的前一次更新向时间较新的方向扫描,并记录当前更新的时间。若无新纪录,仍记录当前更新的时间。

信息分析

设计讨论采用关键词搜索的方式甄别。信息分析设有优先级。未指定分析区间时,加载缓存中储存的、指定分析区间的展示信息并渲染。指定分析区间后,从数据库记录中实时进行查询分析,并向前端提交表单以渲染结果。若信息更新,同时按原区间更新展示信息。

五、附录

5.1 ER 图



5.2 数据流图

